Navigator

NAVO MSRC

FALL 2004

News and information from...

The Naval Oceanographic Office Major Shared Resource Center

Report Documentation Page			*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **2004**	2. REPORT TYPE	3. DATES COVERED **00-00-2004 to 00-00-2004**

4. TITLE AND SUBTITLE **NAVO MSRC Navigator. Fall 2004**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Oceanographic Office (NAVO),Major Shared Resource Center (MSRC),1002 Balch Boulevard,Stennis Space Center,MS,39522**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**28**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

The Directors's Corner...

Steve Adamec, NAVO MSRC Director

The last few months have been exciting ones here at the NAVO MSRC. We're completing a sweeping series of computational enhancements, designated as Technology Insertion for FY04 (TI-04), within the MSRC. The centerpieces of TI-04 are the acquisition and installation of two new IBM POWER4+ HPC systems with a total of approximately 3,500 processors and 6.5 terabytes of memory.

The larger of these systems contains almost 3,000 POWER4+ processors, making it one of the most powerful systems ever purchased from IBM. When complete, the TI-04 enhancements will provide almost 24 new teraflops of aggregate peak computing capability with commensurately balanced storage and networking capabilities.

These new systems will permit the HPC Modernization Program (HPCMP) to offer you what is perhaps one of the best performing HPC environments in the world today. This enormous computational capability, coupled with a sustained 13-year NAVOCEANO focus on supporting the largest and most demanding computational applications, will continue to enable unparalleled advances in the DoD science and technology areas served by the HPCMP.

With all this diverse computational capability that's been fielded across multiple Shared Resource Centers (SRCs) by the HPCMP, it has become critically important for us to redouble our efforts in assessing and implementing common user environments, practices, and tools within and across the SRCs.

The coming implementation of Platform Computing's Load Sharing Facility (LSF) job-scheduling environment across the SRCs should clearly

NAVO MSRC Adds Two New IBM POWER4+ HPC Systems

improve usability and promote a more common "look and feel" to the centers. Here at the NAVO MSRC, we're redoubling our efforts to supplement those activities by continuing and strengthening the links to the HPCMP Programming Environment and Training (PET) program.

Enjoy the coming fall and winter seasons. As always, please take every opportunity to let us know how we can better serve you—your feedback is critically important to us and to the HPCMP.

ABOUT THE COVER:

A variety of images taken from the Dynamic Rotorcraft Simulations project. The project is designed to develop the means to accurately predict complex, unsteady rotorcraft flow fields and aircraft performance using high-fidelity Computational Fluid Dynamics (CFD). The centerfold article in this issue focuses on the research undertaken on the aeromechanics phenomena of the V 22 Osprey tilt rotor operations in low speed flight or hovering. (See page 15.)

Contents

NAVOCEANO

Major Shared Resource Center

- One of the largest scientific computing facilities in the world

- 30+ teraflops aggregate computing capability by September 2004

- 1,000+ terabytes of managed storage

- Serves 4,000+ users across the DoD Services & Agencies

- Secure nationwide network connectivity

- World's foremost HPC center in the delivery of HPC service to operational and R&D oceanographic user communities

...for High Performance Computing

MODAS: The Warfighters' View of the Undersea Environment

By Peter J. Washburn, Naval Oceanographic Office Systems Integration Division

Underwater acoustics is one of the means used by navies worldwide to detect and track potentially hostile submarines. However, ocean acoustics are highly irregular due to the spatial and temporal variability of the ocean thermal structure.

If undersea warfighters can determine the thermal structure of the ocean, they can use this knowledge to gain a tactical advantage: a situation analogous to a land-based army seizing the higher ground.

The Modular Ocean Data Assimilation System (MODAS) is the U.S. Navy's principal tool for depicting the three-dimensional thermal and sound speed structure of the ocean. Tactical Decision Aids (TDAs) convert the MODAS depiction into tactical guidance for the Naval warfighter. This article describes MODAS, its features, its products, its versions, and its anticipated role in the future.

MODAS FEATURES

MODAS uses the method of optimum interpolation to map ocean temperature observations onto a grid. No ocean dynamics, e.g., the equations of motion and thermodynamics, are used.

This method begins with a first guess of the ocean temperature field. The first guess field is often a static climatological field value or a previous analysis. (A static climatology divides the ocean into boxes, and the climatological temperature of each box is the average of the temperature data within that box.)

The first guess is subtracted from the observations to produce observed deviations. The analysis temperature at any point is then a weighted average of the nearby deviations. The weighting factors depend on the type of observation, its age, and the distance of the analysis point from the observation.

The ocean temperature observations are most commonly made using a Bathythermograph (BT). The BT is an instrument that drops through the water column measuring temperature and pressure and transmits the data to the surface via a communications wire.
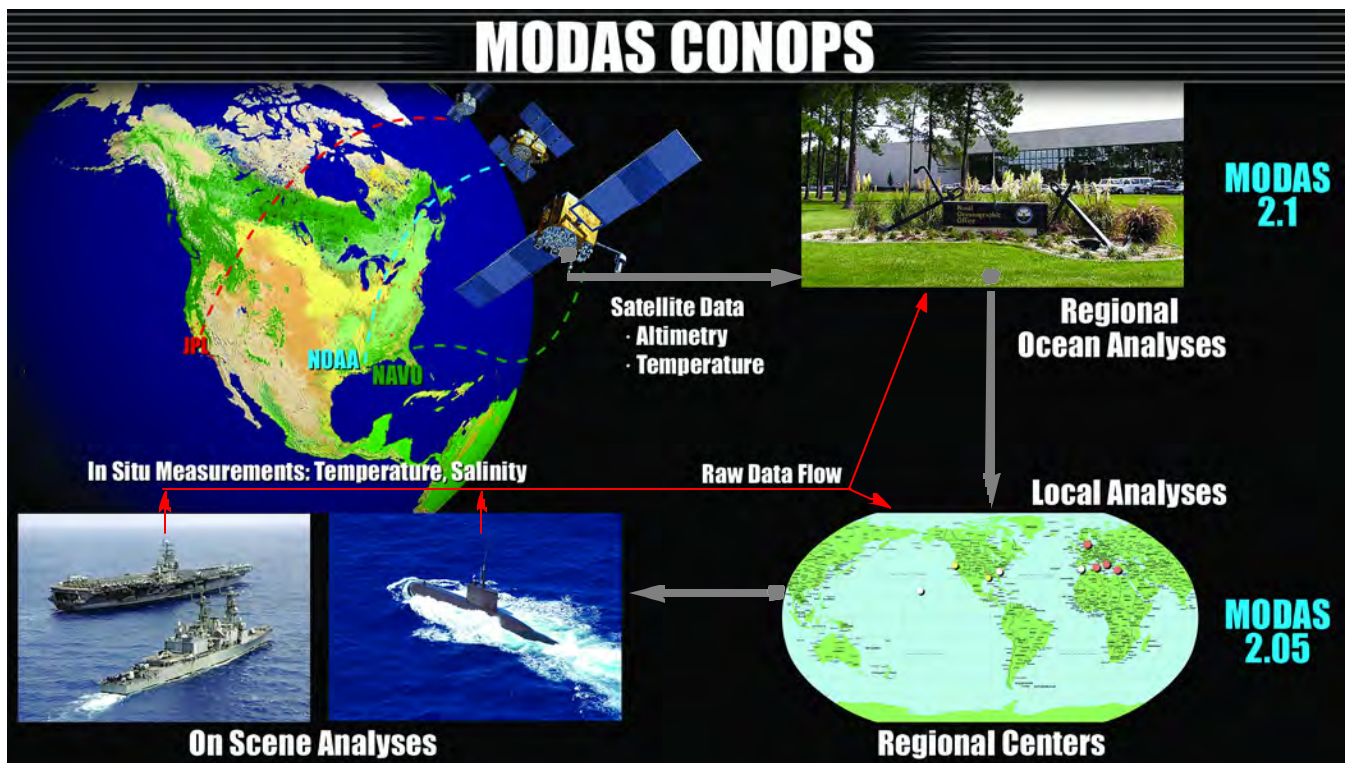
Representation of the Modular Ocean Data Assimilation System (MODAS) Concept of Operations (CONOPS).

However, BT data are very sparse, and the resulting analysis calculates ocean temperatures that strongly resemble the first guesses in areas without BT data. An analysis based on static climatology does not have the sharp temperature gradients characteristic of the real ocean. Hence, the acoustics calculated from these analyses are different from those of the ocean on any particular day.

An innovative feature of MODAS is its use of Dynamic Climatology, which was developed by the Naval Research Laboratory (NRL). A simple way to think about Dynamic Climatology is to consider a mercury thermometer. As the column of mercury warms, the height of the column increases. When the thermometer is calibrated, the user can measure the temperature by measuring the height of the column.

Similarly, as a column of water in the ocean warms, the height of the sea surface increases. Of course, the ocean is more complicated than a mercury thermometer.

In the ocean, salinity has an important effect on sea surface height

changes as does the rise and fall of the thermocline. The thermocline is the part of the water column where the gradient from warm surface water to cold deep water is strongest. This rise and fall is one of the largest factors affecting the sea surface height.

In developing Dynamic Climatology, NRL performed a mathematical comparison of the subsurface temperature profile to the sea surface height and temperature and found that in some areas the subsurface temperature profile could be inferred from the surface data.

Dynamic Climatology is a very important breakthrough for ocean temperature modeling because it enables the use of sea surface temperature and sea surface height data from satellites.

In contrast to BT data, sea surface temperature and height data from satellites are much more abundant.

The analysis begins with the mapping of sea surface temperature and sea surface height data onto a grid using MODAS. MODAS then uses this sea

surface temperature and height at each analysis point and infers a subsurface temperature profile, also called a synthetic BT. The field described by these synthetic profiles can be very realistic. When this field is used as a first guess field, the analysis is realistic in areas without BT data. In areas with BT data, the analysis utilizes both the Dynamic Climatology and BT values.

There are, however, some technical challenges. For example, the method works very well in the Gulf Stream and Kuroshio regions but in some remote regions of the world, there are not enough data to calculate the relationship between the synthetic profiles and the sea surface temperature and height. In other areas, the method does not work as well because of the oceanography of the area.

MODAS VERSIONS

During the early development of MODAS, NRL used it to support Naval exercises by implementing MODAS at the Regional Meteorology and Oceanography (METOC) Centers. Since these early trials, MODAS has become a very well accepted product, and its usefulness to the Naval fleet has increased.

As its usefulness increased, MODAS evolved to meet warfighters' needs. As a result, there are now three major versions of MODAS: MODAS Heavy (MODAS 2.1), MODAS 2.05, and MODAS Lite.

MODAS Heavy runs daily for a number of regions on the high-performance computing infrastructure of the Naval Oceanographic Office Major Shared Resource Center (NAVO MSRC). This "heavy" version of MODAS grids the sea surface temperature and height data and exploits Dynamic Climatology to develop its first guess fields.

MODAS Heavy processing is automatically scheduled on the

Sound Speed Cross Section Typical Warm Eddy

Image of a typical warm eddy generated by MODAS.
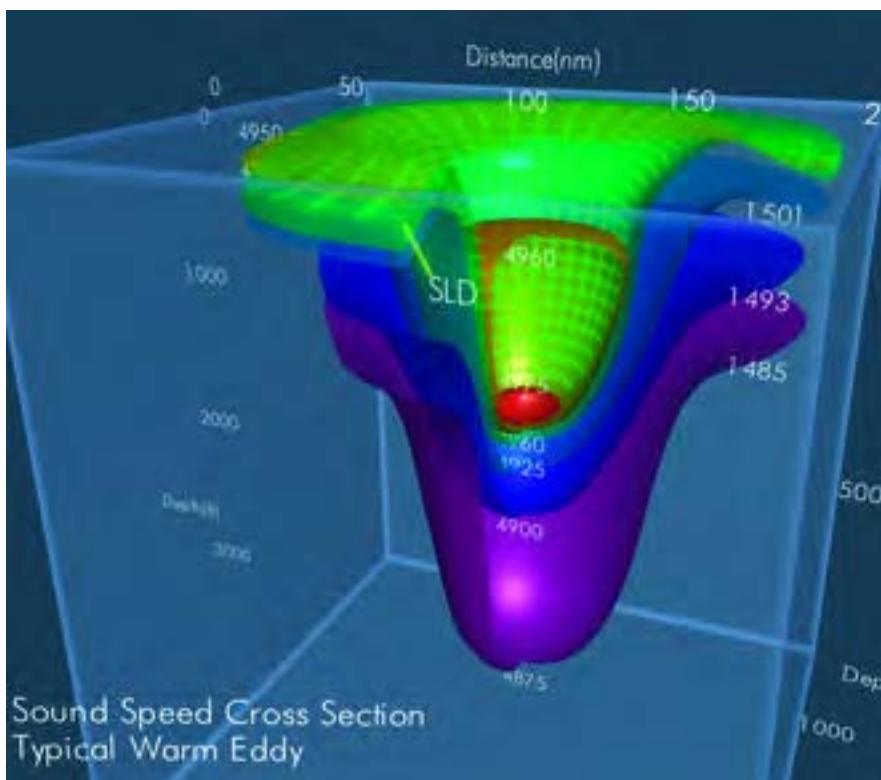
NAVO MSRC supercomputer and managed by the NAVOCEANO Modeling Division (N33).

The intermediate version of the system is MODAS 2.05. This version does not grid the sea surface temperature and height data or exploit Dynamic Climatology. Its first guess field is taken from a previous analysis or climatology.

MODAS 2.05 has a Graphical User Interface that enables a user to edit the BT data, run MODAS on demand, and build graphical products. It is hosted on a high-end UNIX workstation at the Regional METOC Centers.

Finally, there is MODAS Lite. This third version accepts a first guess field (either a static climatological field or a previous analysis) and updates it with BT data. MODAS Lite runs on a Personal Computer (PC) and, like MODAS 2.05, does not exploit Dynamic Climatology.

CONCEPT OF OPERATIONS

Under the MODAS Concept of Operations, NAVOCEANO receives and processes satellite and BT data and generates an ocean depiction using MODAS Heavy. The resulting analyses are sent to the Regional METOC Centers.

The Regional Centers update these analyses using the MODAS 2.05 tool and recent BT data that might not have reached NAVOCEANO. These updated analyses are then delivered to customers at sea.

MODAS Lite enables these customers to update the MODAS fields again using any on-scene BTs. MODAS Lite is integrated with the PC-Interactive Multi-sensor Analysis Training (PC-IMAT) system. PC-IMAT was originally an aid for training the acoustics of the ocean. However, it has been so well accepted that it has become a TDA.

The implementation of MODAS 2.05 at several Regional METOC Centers

presents a life-cycle management challenge. As MODAS is installed at a number of Regional METOC Centers, configuration management is vital. A MODAS defect corrected at one center also needs to be corrected at the other centers.

If the MODAS configurations at the centers diverge over time, a MODAS correction would need to be custom developed for each center. This would dramatically increase the time required to implement MODAS corrections or improvements. Furthermore, the transfer of personnel among centers is more transparent when the centers share common procedures and configurations.

The Systems Integration Division of NAVOCEANO delivers the critical MODAS life-cycle management to the centers via a systems engineering approach.

THE FUTURE

The Navy is transforming from a platform-centric organization to a network-centric organization. The World War II battleship, for example, is platform-centric. It is self-contained

and shoots what it sees.

In a network-centric organization, information becomes the key enabler. Weapons, surveillance systems, tactical displays, and TDAs are on a common network. The fusion of all available surveillance data from across the network leads to a comprehensive awareness of the battlespace, known as Dominant Battlespace Awareness (DBA).

With DBA, a combatant can then shoot not just what it sees, but what it can sense from the data provided by the assets on the network. The network also shortens the cycle time from sensing the target to putting a weapon on that target.

FORCEnet is the network that will enable a network-centric Navy. A key aspect of the network-centric Navy is the implementation of Naval data and applications as ubiquitous Web services.

At present, many of the Navy's applications are unique, one-of-a-kind systems with limited interoperability.
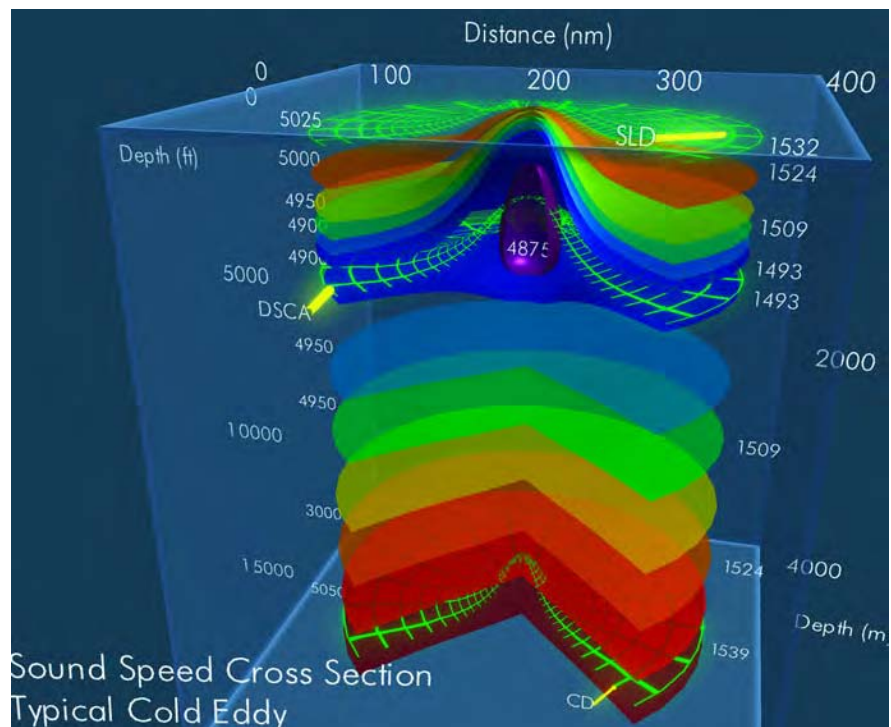
The MODAS system, for example, has

Image of a typical cold eddy generated by MODAS.

many components that are hardwired together. With the current MODAS architecture, customizing the visualization module or other major MODAS components would be a major programming effort.

However, in the Web services approach, each component and data source is offered as a service on the network. The user mixes and matches those services and integrates them using a Web browser on a local computer, which allows for maximum customization.

The use of Web services will be a paradigm shift in the integration of various functions and data in a software application. The FORCEnet vision refers to this customization as composeable services.

NAVOCEANO has a comprehensive initiative to offer its data and applications as Web services. As part of this initiative, the Global Ocean Data Environmental Support Services (GODESS) project will implement a Web-enabled MODAS.

Web services will ensure that NAVOCEANO-provided environmental data and tools such as MODAS will continue to be available to the warfighter as the Navy makes the transformation to network-centric warfare.

CONCLUSION

MODAS is the principal U.S. Navy tool for depicting the ocean thermal structure that is so critical to effective exploitation of ocean acoustics to detect and track submarines.

In response to the needs of the warfighter, MODAS has evolved into three versions, implemented on high-performance computers at NAVOCEANO, on UNIX workstations at the Regional METOC Centers, and on PCs in the fleet.

Currently, MODAS provides a means to augment the very sparse BT data with synthetic BTs through the use of Dynamic Climatology.

As the Navy evolves to a network-centric paradigm, interoperability is critical. Delivering NAVOCEANO products as METOC Web services will ensure that environmental data and tools such as MODAS will continue to serve the needs of the warfighter.
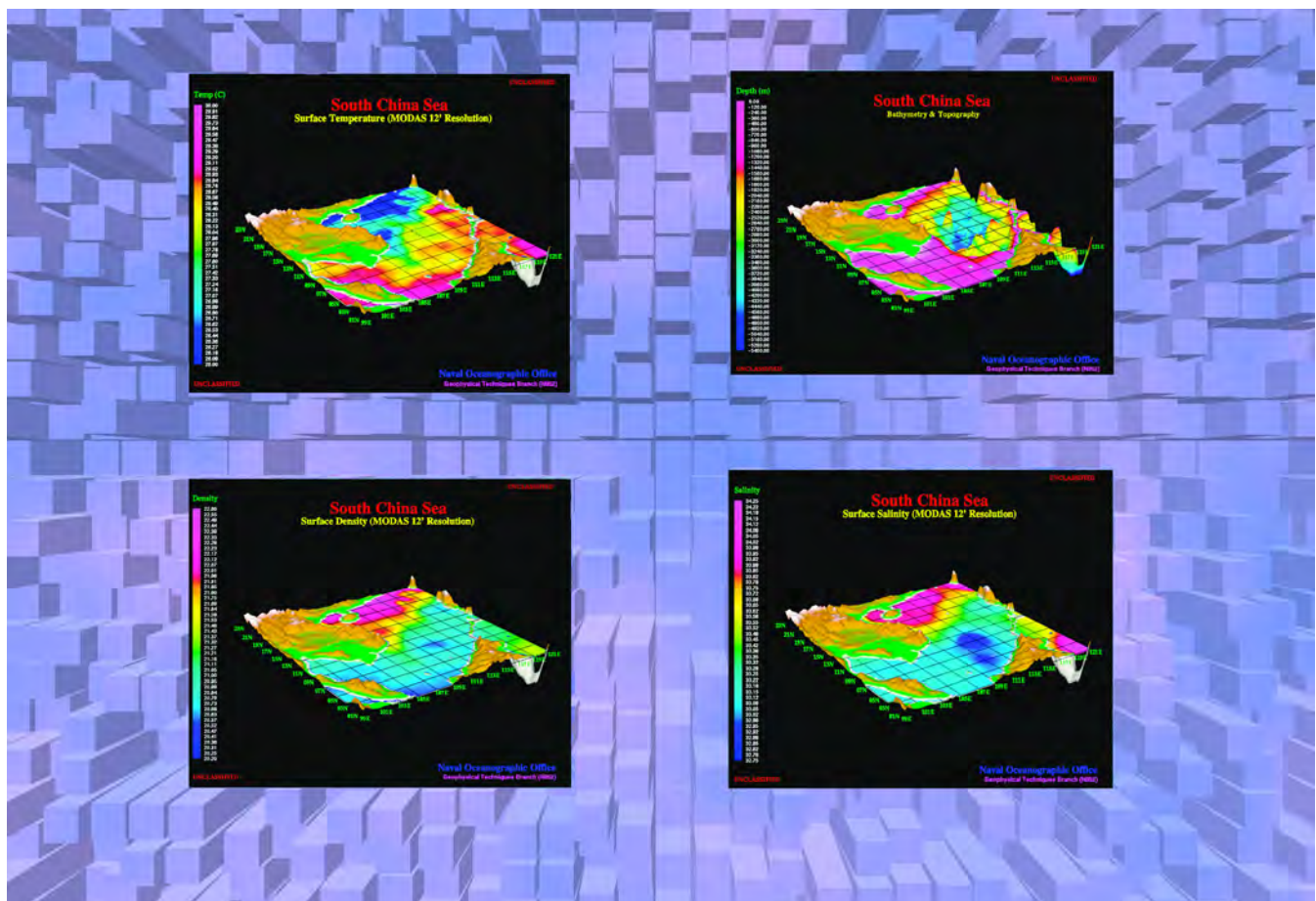
MODAS-generated images portraying the surface density, bathymetry and topography, surface salinity, and surface temperature of the South China Sea.

Visualization Value: Understanding and Developing Smart Materials from First Principles

Sean B. Ziegeler, Visualization Software Engineer, NAVO MSRC Visual Analysis and Data Interpretation Center
Valentino R. Cooper and Andrew M. Rappe, Department of Chemistry and Laboratory for Research on the Structure of Matter, University of Pennsylvania

The development of "smart" materials is of wide-ranging, significant interest to the Department of Defense (DoD). These materials are aptly named as they inherently sense and respond to changes in their environment.

This ability is the primary reason that smart materials show promise in handling the extreme conditions and stringent requirements of military applications. In addition, there are a great many potential applications for this technology, many of which have remained unexplored.

However, smart materials tend to have complex properties that are difficult and expensive to study and experiment with in the real world. Some sort of computational model is preferable to experimental material design because of the reduced cost and time.

To that end, a method of chemical modeling known as first principles is being used to simulate specific behavior and predict suitable compositions of smart materials for future use. Given the various applications of smart materials, the research described in this article has been narrowed to the following two objectives:

1. Study the salient features of chemical processes on metallic surfaces to reveal the behavior of corrosion and how smart materials can affect corrosion.

2. Analyze the complexities of piezoelectric ceramics, a type of smart material that can be utilized as sensors in SONAR devices.

These two objectives have significant relevance to several branches of the DoD.

The first research objective–in which a smart material could be chemically combined with or used as a coating for a metallic surface and its properties might allow it to prevent or reduce corrosion–is of particular interest to the United States Navy and Air Force. Both services are engaged in a continual search for new means to reduce maintenance and extend the lifetime of ships, planes, and other vehicles and equipment by limiting the effects of corrosion. The second research objective is of particular interest to the Navy–it could lead to improvements in the understanding of piezoelectric ceramics, which would contribute to the development of next-generation SONAR. Both lines of research are being conducted at the University of Pennsylvania (UPenn) as part of a Naval Oceanographic Office Major Shared Resource Center (NAVO MSRC) Challenge Project.

The researchers have developed a first principles model using a combination of several well-known methods and novel techniques that is both flexible and powerful. Truly understanding the output of such a complex model requires visualization to interpret the large volume of data and discern the potential intricacies within that data. The NAVO MSRC Visual Analysis and Data Interpretation Center (VADIC) provides assistance with the development of visualization applications for this research effort. This article describes how the model output is transformed so it can be visualized, how it was integrated into the software, and the techniques that were used for rendering the visualizations.
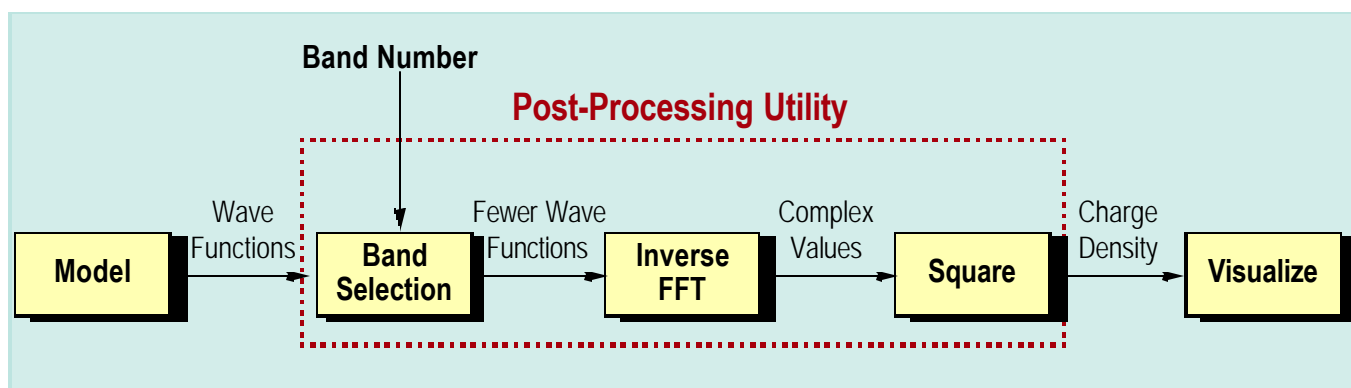
Figure 1. Step-by-step diagram of the Post-Processing Utility.

FIRST PRINCIPLES AND QUANTUM SIMULATIONS

A first principles simulation refers to predicting the behavior of a system from a governing equation without any experimental inputs or fitting.

In this project, the researchers begin with the Schrodinger equation and model the behavior of atoms and electrons starting at the quantum mechanical level. This is a difficult problem computationally and requires the solution of thousands of simultaneous equations.

The model discussed in this article uses Density Functional Theory (DFT). DFT is an accurate and efficient method of simulating electron interactions that uses electron charge density as the basic variable for systems of equations. This is faster than traditional quantum simulations.

The model also uses the Fast Fourier Transform (FFT) to operate in both real and reciprocal spaces, which allows the use of the most efficient computations for a given space.

The direct output of the model is a set of wave functions, given as complex numbers in the reciprocal space. A post-processing utility must be used on this output to produce the desired data for visualization.

A smaller group of the wave functions is culled from the output, based on the electron's energy, enabling the user to focus on electrons participating in important chemical or physical processes. An inverse FFT is performed on this smaller group of wave functions, effectively transforming the data from reciprocal space to real space.

The data points produced by the inverse FFT are still complex numbers, but when squared, produce real numbers that correspond with electron charge density. Figure 1 illustrates this process.

VISUALIZATION SOFTWARE

The VADIC staff is developing a visualization application to examine charge density three-dimensionally. A standard, open-source software toolkit, OpenDX, is used to implement the visualization application.

OpenDX is a flexible software package allowing a wide range of graphics techniques, image processing tools, and visualization methods which are provided as modules in a visual programming environment. Each module performs some operation, such as importing data, processing data, producing visual output, rendering isosurfaces, etc.

These modules are connected together into what is known as a visual program. The connections are made graphically in OpenDX's visual programming editor. Figure 2 is an example of a simple visual program that: (1) imports data; (2) creates an isosurface; (3) colors the isosurface; and (4) displays the results on the screen. Figure 3 is a more complete example that shows parts of the visual program used specifically for this project.

One attractive feature of OpenDX is that it also provides modules that implement user interface components. These modules are placed within the visual program, but also have an interface object that allows the user to type in or select values. The selected values are passed from the output tab of the interface module to the input tabs of connected modules. Groups of interface objects can be arranged together into control panels.
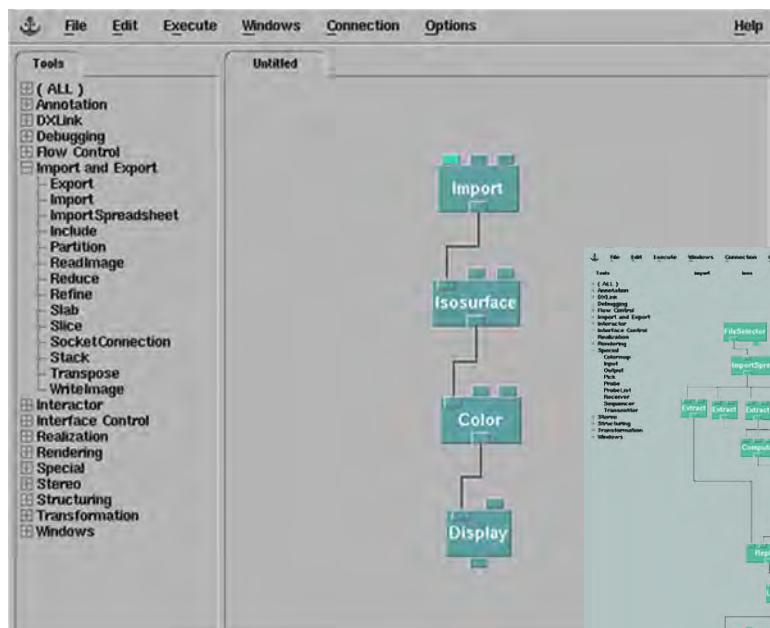

Figure 2. (Above) A simple example of an OpenDX visual program.
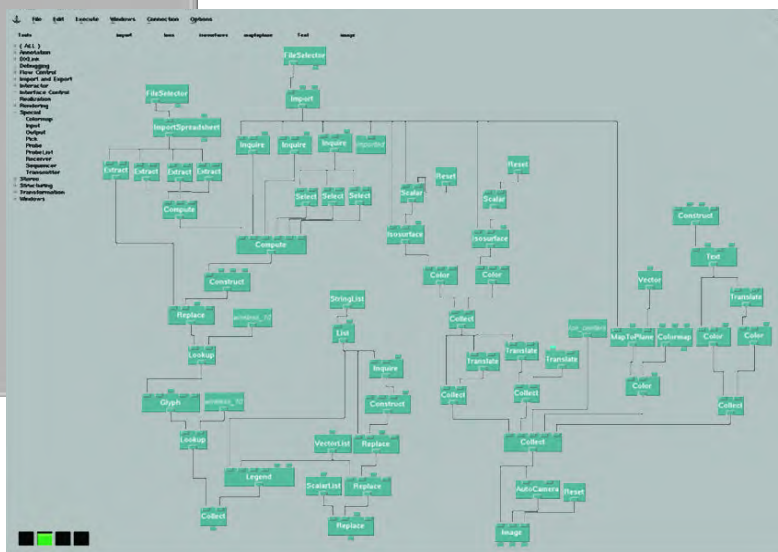
Figure 3. (Right) A more complex example of a visual program.

This allows developers, such as the VADIC staff, to build entire interface systems providing significantly more ease of use for the investigator. Figure 4 illustrates two user interface control panels with embedded interface objects.

VISUALIZATION METHODS AND TECHNIQUES

In the case of the UPenn research, one of the investigator's primary interests was to integrate the post-processing utility directly into the OpenDX visual program. This would result in a seamless visualization system that would work directly with the output of the model.

In addition, an interface object could be added to allow the investigator to select the desired band number for the post-processing utility. The post-processing utility was written by the investigator in Fortran90, but since it was compiled as an executable, OpenDX has the capability to execute it as an external program and "pipe" in the program's output. Furthermore, parameters such as file names and band number can be passed to the external program.

For this model, the investigator wanted to use some form of volume visualization to see specific values of charge density over the entire spatial volume of data. Given that requirement, the VADIC staff determined that semi-transparent isosurfaces are an excellent fit for this specific application.

An isosurface is a surface object that is rendered everywhere the data set is equal to a specific value. Figure 5 shows two isosurfaces of specific values, each transparent and with a different color.

The mere isosurfaces are not enough, however. It is helpful for the UPenn researcher to have contextual data in the form of ion center points and sizes. This provides the ability to correlate behavior and features in the data with the locations of certain types of ions. Types, sizes, and a color code for ions can be added via the user interface (the left control panel in Figure 4 shows the ion type lists).

The list of actual ion types and positions is provided by the user as a separate data file. This data file is imported by the OpenDX visual program and

manipulated automatically to align with the data set.

The ions are rendered as spheres which are scaled in proportion to the atomic size of the specific type of ion. For further information, a legend can be displayed, indicating which color of sphere represents a type of ion. Figure 6 is an example of one data set with two isosurfaces and a set of corresponding ions.

RESEARCHER ANALYSIS

To understand how these visualizations benefit this smart materials research, it is necessary to consider some of the science that is present behind the scenes. The corrosion processes of Objective 1 occur through a sequence of steps:

1. A molecule, like oxygen, is adsorbed to a surface.

2. This adsorbed molecule disassociates into something more reactive.

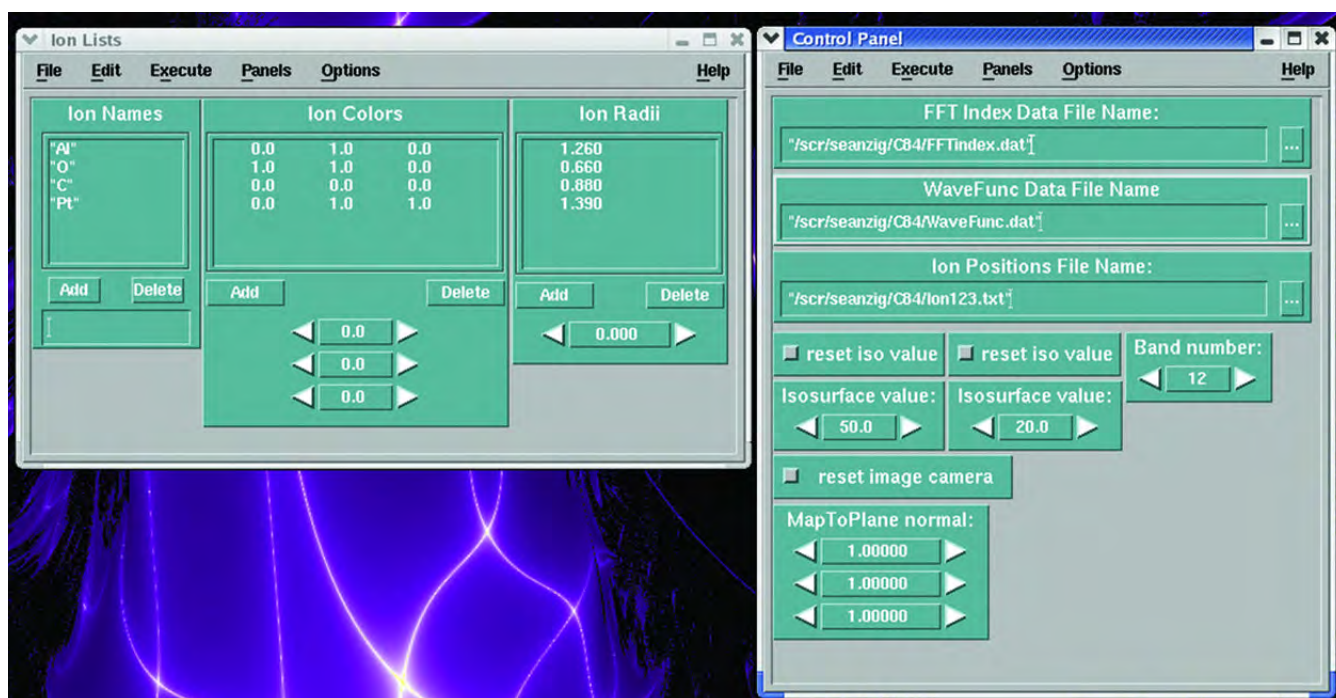3. It then interacts with the surface, resulting in corrosion.

Figure 4. Control panels with interface objects from the visual program in Figure 3.

For Objective 2 the science behind piezoelectric ceramics is in how they respond to changes in their environment. An external mechanical force (such as a sound wave) can deform the crystals, thereby inducing an electric field within the piezoelectric ceramic.

The field can be monitored, resulting in information about the mechanical force that was observed. Conversely, an electric field can be applied to some other type of piezoelectric ceramic, causing it to deform and produce sound waves.

The above-mentioned processes involve the movement of electrons, i.e., a change in the charge density within these materials. An understanding of these fluctuations of the charge density would result in an understanding of either how the adsorption of a molecule occurs (helping with Objective 1) or how a change in piezoelectric response occurs (helping with Objective 2).

Observing charge distribution in such a way can give us a window into how smart materials respond to their environment and what modifications to the material will result in the desired behavior for the chosen application.

With charge density fluctuations in mind, it is often necessary to look at the induced charge density of a material. The induced charge density is simply the change in charge density of a material as a result of altering some external variable.

For example, the induced charge density of a metal surface after the adsorption of a molecule such as oxygen will provide information as to which metal electron orbitals are most involved with the bonding of this molecule.

Once a clear understanding of how oxygen molecules bind to the surface is obtained, it is possible to think of ways to modify the particular orbitals involved in the binding of oxygen to the surface. By weakening these interactions it will be possible to slow down the corrosion processes within these metals.

The isosurfaces of induced charge density displayed in Figures 5 and 6 show specific values of interest to the UPenn investigator. Given renderings such as those, it is possible for a researcher to investigate the bonding of oxygen molecules to metallic surfaces. Additionally, the invaluable contextual information, i.e., the ion center points and sizes, allows the
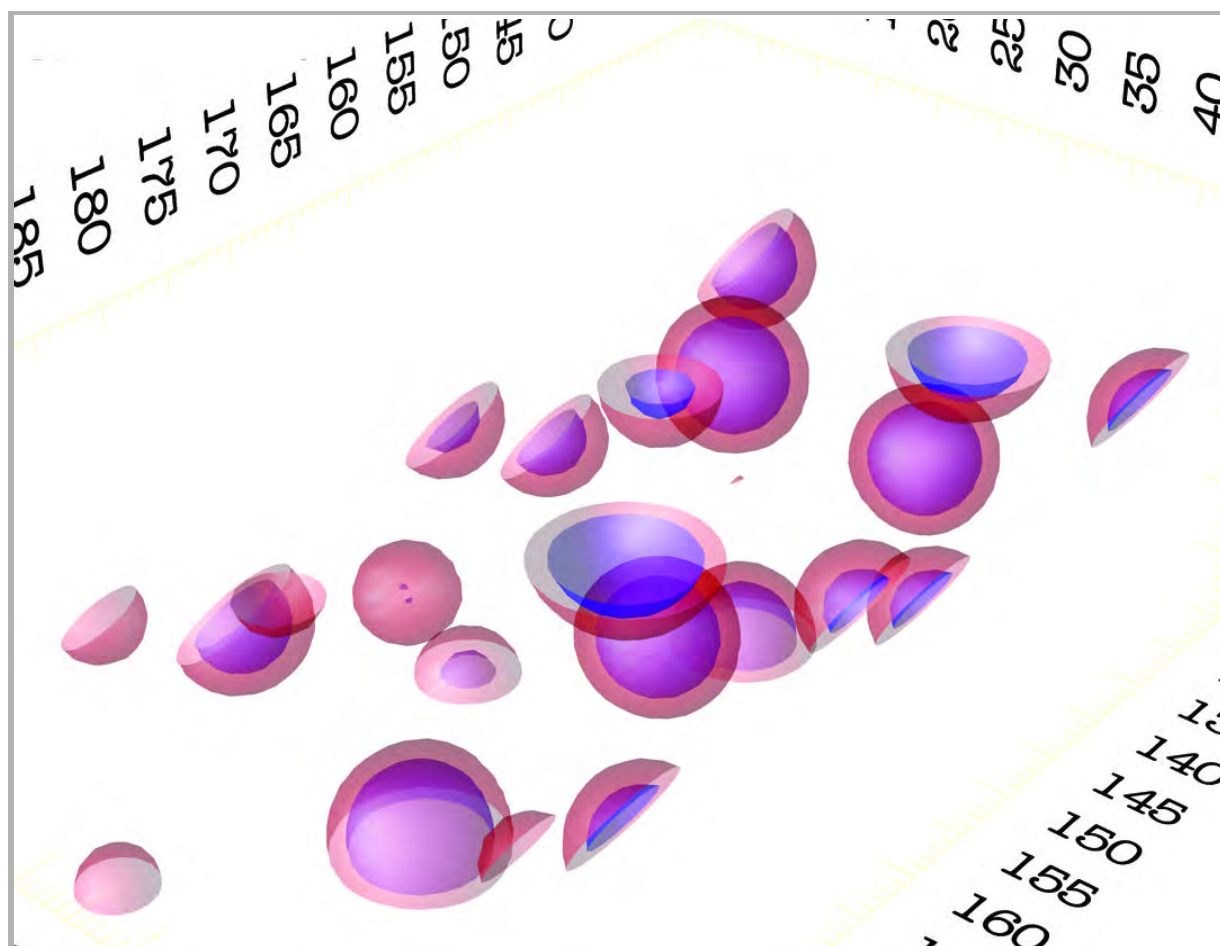

Figure 5. Isosurfaces of value 20 (red) and 50 (blue) respectively.

UPenn investigator to notice where certain interesting features in the induced charge density occur with respect to the oxygen molecules and metal ions.

Unfortunately, an isosurface rendering of induced charge density may not be enough. An example is in the case of oxide supported metal thin film surfaces.

Traditional isosurfaces of induced charge density would compare changes in the charge density of metal atoms in the thin film metal layer (supported on the oxide surface) with that of a block of metal (with no oxide support).

Changes in charge densities in this case would relate to a metal suspended in a vacuum, and not to the block of metal, therefore leaving researchers with misleading information.

For this reason, using band population charge densities is a more useful tool. Each band corresponds to a particular electron orbital within the material. By mapping out these bands, the UPenn investigators expect that it will be possible to look at the charge density changes within a particular orbital and thereby be able to quantify the changes in the orbital as a result of changes in the external environment.

In this is seen the necessity of the post-processing utility discussed previously and shown in Figure 1. Selecting the band number within the post-processing utility should effectively select, from within the data, the induced charge density of a given electron orbital that is of interest to the investigator.

CONCLUSION

VADIC has provided a unified visualization system constructed within a standard framework.

This system shows a specific variable of interest, induced charge density, with multiple, specific values of interest using semitransparent isosurfaces. Furthermore, integrating the post-processing utility directly into the visualization application creates a system capable of even more flexible and powerful exploration of the data.

This ability has new implications for being able to accurately map out the flow of electrons within a material. The visualization system allows the researchers to investigate these changes, providing new insights into how to design more efficient smart materials.

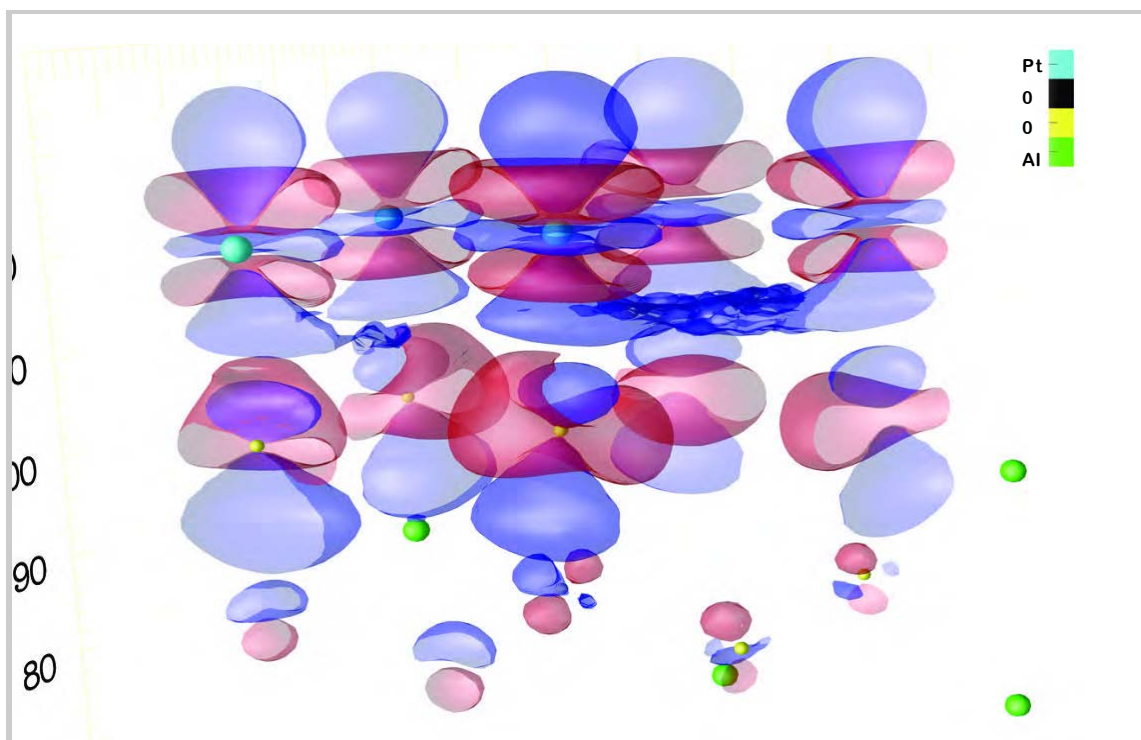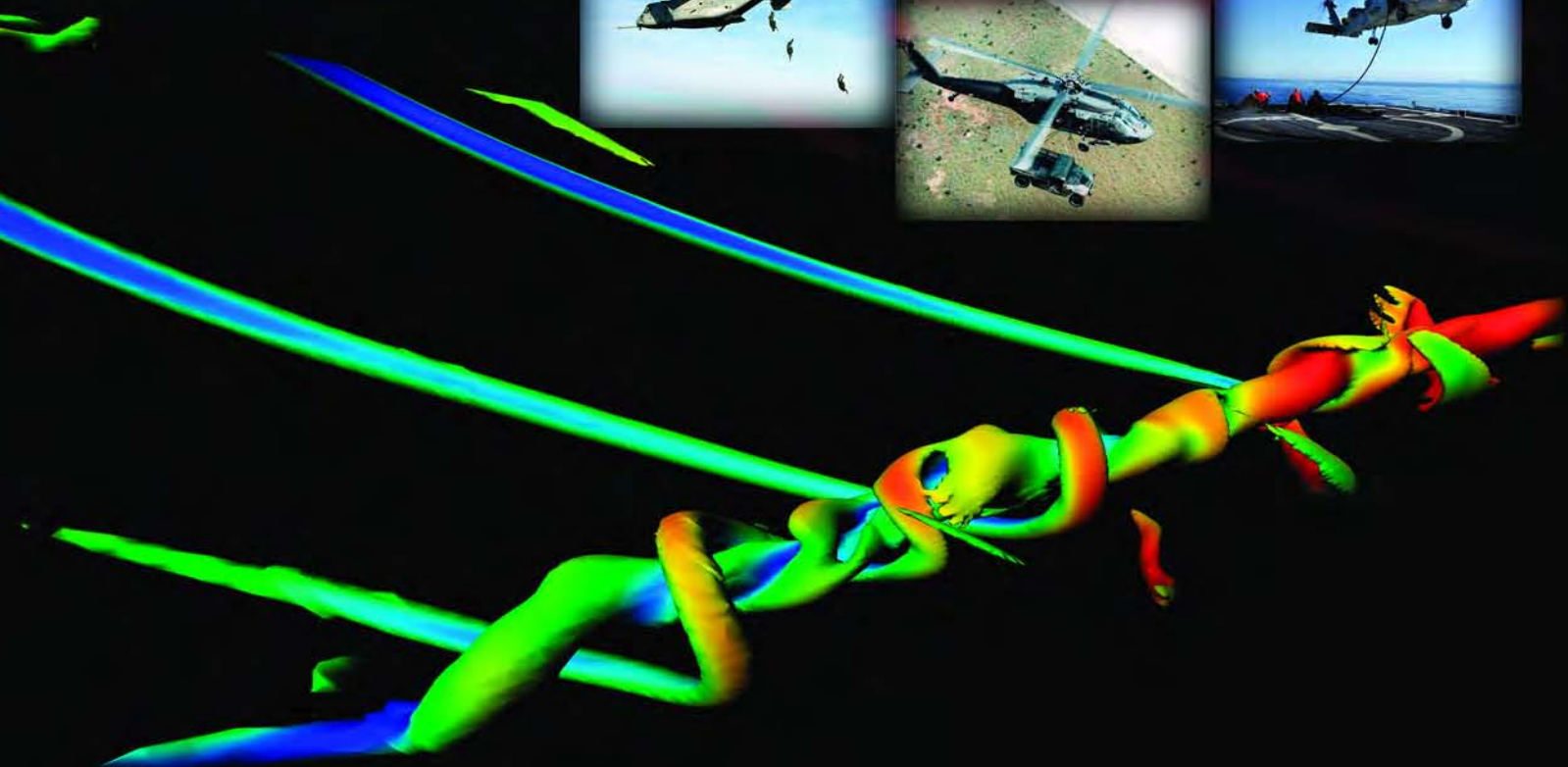**Figure 6. Isosurfaces of value -13.7 (blue) and 17.0 (red) with ions displayed as scaled spheres.**

Tilt Rotor Aeromechanics Phenomena in Low Speed Flight

Mark Potsdam
Army/NASA Rotorcraft Division
U.S. Army Aeroflightdynamics
Directorate (AMRDEC)

Tilt rotor aircraft are recognized for their significant potential impact on both military and civilian aviation. In them, the range and speed of a turboprop airplane is augmented by the ability to operate from confined areas like a helicopter. The helicopter mode capability allows tilt rotors to hover, maneuver around an airfield at low speed, and conduct shipboard operations with wind over the deck. The objective of the work described in this article is to investigate important aeromechanics phenomena affecting the V 22 Osprey tilt rotor in operations requiring low speed flight in any direction or hovering in wind conditions. During V 22 critical azimuth flight testing (See Figure 1) designed to evaluate control margins and pilot workload under such flight conditions, aeromechanics phenomena were identified that adversely impacted handling qualities.[1] Independently, the phenomena were recognized in Computational Fluid Dynamics (CFD) calculations that have been particularly useful in detailing the causes of the phenomena.[2]

Article Continues Next Page

COMPUTATIONAL MODELING

A high fidelity model for CFD analysis was constructed of a V 22 configuration in hover. The CFD calculations used the Reynolds-averaged Navier-Stokes code OVERFLOW-D,[3] which has been developed by National Aeronautics and Space Administration (NASA) and the U.S. Army under a Department of Defense (DoD) Common High Performance Computing Software Support Initiative (CHSSI) project and applied to a wide range of fluid dynamics problems.

OVERFLOW-D includes capability for time-dependent, rigid body motion of components such as individual moving rotor blades. Solutions are computed on structured, overset grids using body-conforming "near-body" grids and Cartesian "off-body" grids in the wake. The Baldwin-Barth turbulence model is employed. Figure 2 shows the V 22 near-body surface grids. The total number of grid points in all 410 grids is 47.6 million.

Solutions are computed on parallel supercomputers or workstations communicating using the Message Passing Interface (MPI) protocol.

Both the overset domain connectivity and flow solver modules have been parallelized for efficient, scalable computations. Scalability of OVERFLOW-D on the V 22 grid is illustrated in Figure 3 as speed up from a baseline, defined as 32 processors, and compared with ideal linear speed up.

Due to efficient memory usage, up to 2 million grid points can be placed in 1 gigabyte (GB) of memory, creating a lower limit of approximately 32 processors for this problem. Parallel efficiency typically falls off when the number of grid points per processor falls below 250,000, creating an upper limit of approximately 192 processors.

These simulations were run on 128 processors of the Naval Oceanographic Office Major Shared Resource Center (NAVO MSRC) IBM POWER4 (MARCELLUS). Use of 128 processors is seen to be a good compromise between speed (7.6 seconds per step) and efficiency (73 percent). Each rotor revolution requires 5 wallclock hours for 2400 iterations per revolution. OVERFLOW-D simulations were run with the V 22 hovering in a 35-knot wind from 0 (headwind), 45, 90, 135, and 180 (tailwind) degrees wind azimuth. The baseline rotor speed was 413 Revolution Per Minute (RPM) (hover tip Mach number of 0.736), and rotor thrust coefficient was 0.015. Nearly 100,000 Power4 hours were used for the present investigation.
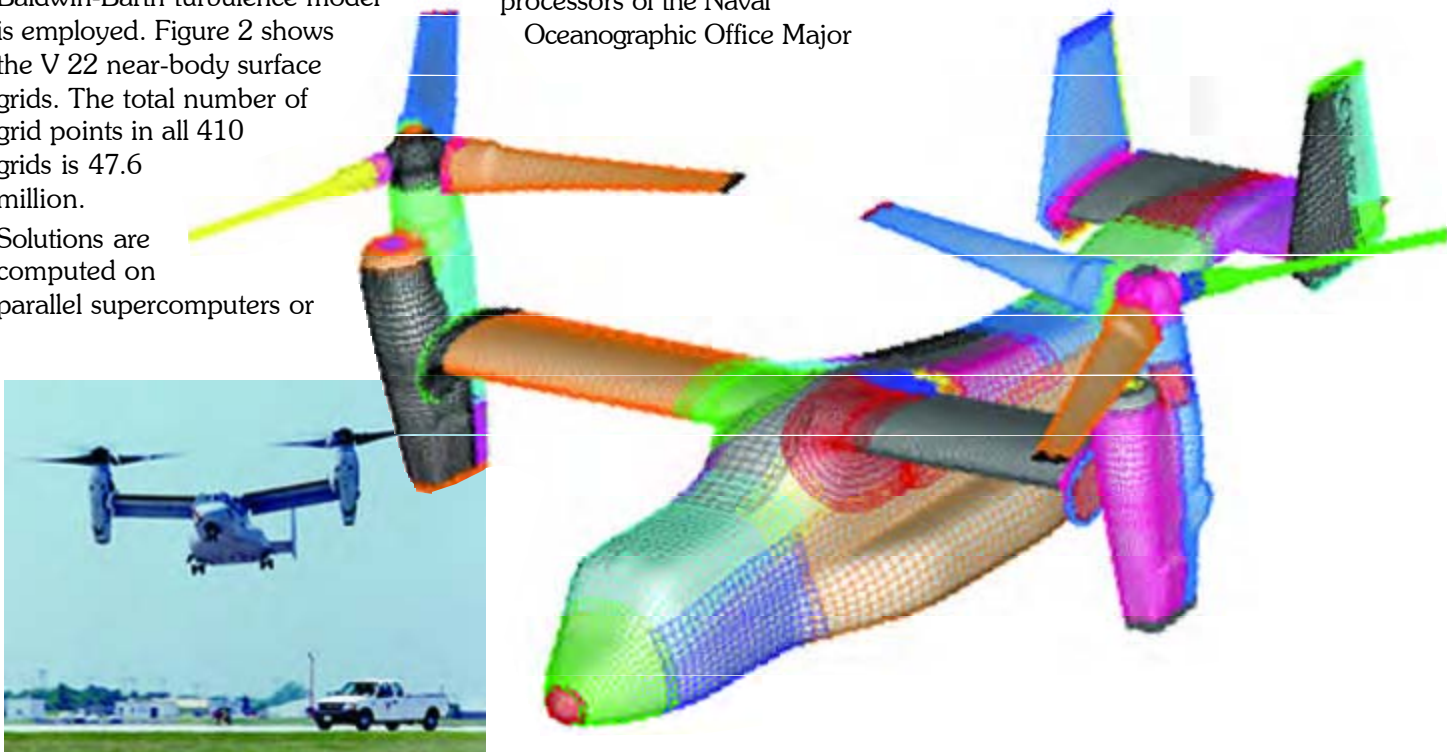

Figure 1 (Left). V 22 Osprey in helicopter mode during critical azimuth flight testing.
Figure 2 (Right). V 22 Osprey near-body surface grids.

TILT ROTOR AEROMECHANICS PHENOMENA

Pitch-Up With Sideslip (PUWSS) is a well-understood aeromechanics phenomenon in which the upwind rotor wake impinges on the horizontal tail, causing the aircraft to pitch up.[1] The phenomenon is most critical in quartering headwinds. In flight, increased fuselage pitch attitude is counteracted by longitudinal stick displacement and forward nacelle tilt. In the CFD calculations, PUWSS is noted as an increase in airframe pitching moment (positive—nose up). An OVERFLOW-D time-averaged pitching moment is shown in Figure 4 for a 35-knot wind speed with and without rotors.

Note that the pitch-up is not indicated in the fuselage-only calculations, indicating that this is an adverse rotor-airframe interaction. Calculated pitch-up trends are in excellent agreement with flight test observations.

The flow visualization in Figure 5 confirms the rotor wake impingement on the empennage. In this image, extracted from a time-dependent animation,

particles released from the upwind (left) blade tips impact the aft fuselage and empennage. The majority of the pitching moment comes from the tail and fuselage contributions. Pressure forces in the download direction, shown in Figure 6 as a function of wind azimuth, are a further indication of the decreased lift on the tail and aft cargo ramp. Blue coloring on the configuration indicates upload, while red indicates download. Comparing the 0° headwind and 90° sidewind cases with the 45° quartering headwind, reduced upload on the tail upper surface, and increased download on the tail lower surface and fuselage underside translate into the pitch-up.

Increased power requirements in sideward flight were also first revealed in V 22 critical azimuth flight testing.[1] The power required to hover in sidewinds is 10-20 percent higher than no/low-wind hover. In constant high wind conditions the power required to hover increases drastically (up to 80 percent) as the wind direction moves from a headwind toward a sidewind. The increase in power required in sideward flight can be directly correlated with an increase in airframe download. CFD-predicted download trends shown in Figure 4 indicate increasing download as wind azimuth is increased up to the sidewind condition. The unpowered airframe shows a trend of increased download with sidewind, but this phenomenon is magnified by the rotor flowfield. The consequences of the rotor-fuselage interaction are significant. CFD excels here because developing an experimental model that can be turned through 180° of sideslip without interfering with the fuselage flowfield is difficult.
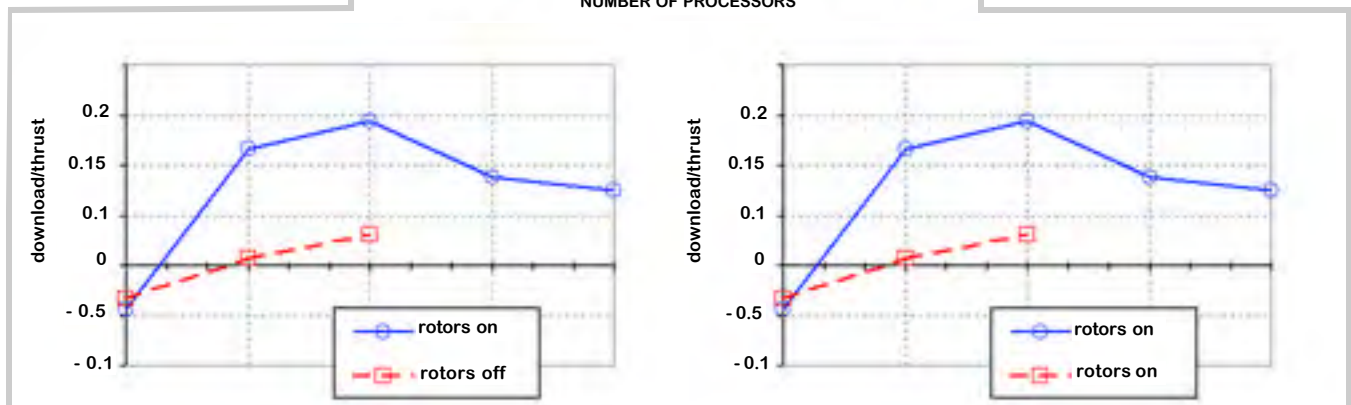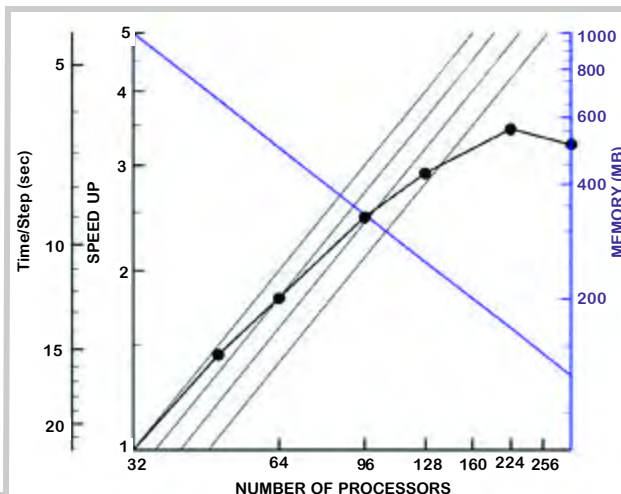
Figure 3. (Top) Illustration of the scalability of OVERFLOW-D.
Figure 4. (Bottom) OVERFLOW-D time-averaged pitching moment for a 35-knot wind speed with and without rotors.

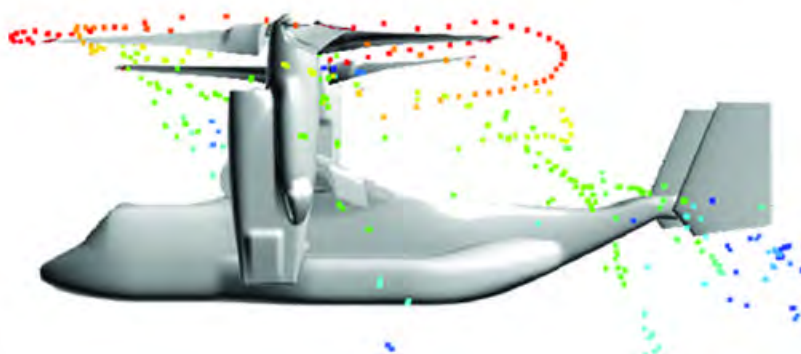**Figure 5. Flow visualization of wake impingement on the empennage.**

Pressure forces in the download direction (See Figure 6) indicate that in a 35-knot headwind all parts of the airframe are lifting. As the wind passes through 45°, the upload on the wing and aft fuselage upper surfaces is reduced. At 90° the download due to the rotor wake on the wing is significant. The flat bottom lower surface of the fuselage also has a large influence. The suction on this surface steadily increases as the wind direction approaches 90°.

CONCLUSIONS

Aeromechanics phenomena affecting the V 22 tilt rotor in low speed flight or while hovering in wind from varying azimuths have been successfully investigated using CFD. Results compare well with V 22 flight characteristics. CFD clearly elucidates the important rotor-fuselage interactions involved, and, therefore, the necessity of complete configuration modeling including the rotors. From these and other CFD calculations on V 22 hover and cruise configurations, the V 22 program has gained significant insight into tilt rotor interactional aerodynamics. Calculations have shown elevator deflection to effectively reduce the pitch-up and suggested the flight testing of an open cargo door for download alleviation. Similar simulations can provide a foundation for future tilt rotor design and development and risk reduction in flight test. DoD high performance resources at the NAVO MSRC were instrumental in performing these large-scale, high-fidelity rotor-fuselage calculations.

0 Degrees

45 Degrees

90 Degrees

Figure 6. Pressure forces in the download direction.

References

1. Muldoon, F., "Numerical Methods for the Unsteady Incompressible Navier-Stokes Equations and Their Application to the Direct Numerical Simulation of Turbulent Flows," Ph.D. thesis, Louisiana State University, 2004.

2. Narayanan, S., P. Barooah, et al., "The Dynamics and Control of an Isolated Jet in Cross Flow," AIAA Journal, 2003.

3. Smith, S.H. and M.G. Mungal, "Mixing, Structure, and Scaling of the Jet in Crossflow," Journal of Fluid Mechanics 357: 83-122, 1998.

THE PORTHOLE

Naval Oceanographic Office

Steve Adamec, Director, NAVO MSRC; RMDL(S) McGee, NAVOCEANO; Command Master Chief Johnstone, NAVOCEANO; CAPT Philip Renaud, NAVOCEANO; ADM Fallon, Commander, Fleet Forces Command.

Steve Adamec; Frank Martin, NASA Information Assurance Officer, Terry Bordelon, Manager, NASA Stennis Telecommunications; Pat Dunnington, NASA Chief Information Officer; David Throckmorton, Deputy Center Director, NASA Stennis.

Steve Adamec; CAPT Rob Lawson, Deputy Oceanographer of the Navy; BGen Thomas E. Stickford, Commander, Air Force Weather Agency (AFWA); RADM S. Tomaszeski, Oceanographer of the Navy and ForceNet; MSgt Lee, AFWA; Command Master Chief Johnstone.

Visiting delegation from the National Geospatial-Intelligence Agency.

Sterling Barbour; Reeves Barbour; Mississippi First Lady Marsha Barbour; Steve Adamec, Director, NAVO MSRC.

Visit of the Department of the Navy, Chief Information Officer (DONCIO) Policy Group on Open Source Software.

Naval Oceanographic Office

Dave Cole, NAVO MSRC; Christine Jarrett, NAVOCEANO; Bruce Vogt and Dr. Cynthia Decker, Intelligence, Surveillance, and Reconnaissance Division, ForceNet/ Oceanographer of the Navy.

The Change of Command Ceremony and picnic...
Farewell to CAPT Lumpkin...Welcome Aboard CAPT Best!

NAVAL OCEANOGRAPHIC OFFICE

NAVOCEANO Command Picnic 2004—fun and games for everyone.

NAVO MSRC PET Update

Eleanor Schroeder, NAVO MSRC Programming Environment and Training Program (PET) Government Lead

PET is almost halfway through Year 4. The summer flew by with help from our annual Summer Intern Program. This year, we had four students participate here at the Naval Oceanographic Office Major Shared Resource Center (NAVO MSRC) and one who was assigned to the Naval Research Laboratory (NRL), Stennis. The five students were Tiffany Broadnax (Jackson State University), Fiseha Negasi (Central State), Benjamin Payment (Mississippi State), Allison Scogin (Mississippi State), and Darrin Woods (Central State). An article about one section of the intern program follows this update.

This might be a good time to focus on some of PET's successes. Component 1 concentrates its efforts on the Climate/Weather/Ocean Modeling and Simulation (CWO) and Environmental Quality Modeling and Simulation (EQM) communities, as well as provides assistance in areas relating to the Computational Environment (CE). The CWO, EQM, and CE teams have significantly impacted their respective DoD user communities through personal contact and support by the onsite personnel, the Functional Area Point of Contacts (FAPOCs), and through project efforts.

In EQM, the optimization of CH3D-SED yielded a significant improvement in performance and decreased the time for sediment calculations by nearly 40%. Enhancements to CE-QUAL-ICM included the migration of transport algorithms such as discontinuous Galerkin and the Normalized Variable Diagram into the latest version. The biogeochemical module, TRCHEM, was provided for coupling into Advanced Development Hardware (ADH). These EQM efforts are providing direct benefit to the Army Corps of Engineers missions, such as support for military logistics across the shore and water quality assessments.

In CWO, assistance was provided for coupling ocean-acoustic dynamics, using the Eulerian/semi-Lagrangian numerical model for fluids (EULAG) and Interactive Groundwater (IGW) models, which enabled the new project to effectively use NAVO MSRC resources and accomplish its research goals.

Support to the Nearshore Wave and Circulation (SWAN) model included enabling high-resolution nearshore wave modeling experiments and accelerating the transition of the advanced shallow water wave model to operations. Assistance with the circulation modeling in coastal regions resulted in the official release of the Message Passing Interface (MPI) parallel code ADCIRC 2-D/3-D V43.03. High Performance Computing (HPC) portability, extensibility, and performance improvement were enabled in the NRL Spectral Element Atmospheric Model (NSEAM) code.

Assistance in these areas provide direct benefit to the operational forecast support activities which in turn support worldwide Department of Defense (DoD) military operations, such as Iraqi Freedom, Enduring Freedom, and associated exercises.

Provision by the CE team of memory debugging tools, such as Valgrind, enabled users to find memory leaks that they didn't know existed. Continued support to the High Performance Computing Modernization Program (HPCMP) Benchmarking Team was offered via the use of Performance Application Programming Interface (PAPI) and Tuning and Analysis Utilities (TAU) to collect Level 1 profiling data for CWO and Signal and Image Processing (SIP) benchmark codes and providing a recipe for using TAU to collect Level 2 profiling data.

The Consistent Well-Documented Computational Environment successfully completed platform documentation for the IBM POWER3/4, SGI Origin 3000 series, and the HP AlphaServer platforms. CE efforts have benefits across all DoD programs in that improved debugging and performance analysis tools can greatly enhance the productivity of DoD software systems and applications.

More information about these efforts and efforts by the other components can be found on the PET Online Knowledge Center (OKC), https://okc.erdc.hpc.mil/index.jsp.

And what about strategies for the future? The Joint Technology Master Plan (JTMP) looks at accessible technologies as they apply to identified and traceable DoD HPC user requirements. The process allows for strategic planning that supports both core and Project Task activity in each functional area as well as across the functional areas.

The JTMP captures evolving DoD HPC user requirements from all the PET functional areas to ensure DoD HPC users can access and implement those new technologies that will benefit the warfighters. This set of requirements will form the basis for future development and selection of PET activities. The intent of the JTMP is to provide PET a rationale for proceeding along paths of technology transfer that are most relevant to critical DoD needs in HPC. This is a massive task requiring the active participation and cooperation of all stakeholders, and updates will occur annually.

NAVO MSRC PET Update
Summer Interns

Tom Cortese, ICL/UTK, PET Computational Environments Onsite

Over the past few months I have had the privilege of working with three undergraduate students as part of the Programming Environment and Training (PET) Summer Intern Program. Each of these students studied several aspects of high-performance computing within the Department of Defense (DoD) environment and gave a final presentation and a written report at the end of the summer. What follows is a summary of their experiences and accomplishments based on their reports and presentations.

All three interns went through several mini-classes on topics such as working in a Kerberos environment, preparing and submitting jobs to a batch scheduler, discretizing a differential equation and solving it numerically, and computer architectures and parallelization with Message Passing Interface (MPI) and OpenMP. In addition, a Mandelbrot Set zoom animator was built in several stages throughout the summer.

The first version of the code ran on one processor and generated one animation frame. OpenMP parallelization was added in order to use multiple processors within a node. MPI parallelization was then added to distribute multiple frames across

The PET Intern Program Team: Seated: Benjamin Payment, Intern, MSU; Fiseha Negasi, Intern, Central State (Ohio); Allison Scogin, Intern, MSU; Tiffany Broadnax, Intern, Jackson State. Standing: Tim Campbell, Asst. Research Professor, MSU; George Heburn, Director PET Component 1 NAVO, MSU; Tom Cortese, ICL/UTK PET Computational Environments Onsite; Darin Woods, Intern, Central State (Ohio).

several nodes using full hybrid OpenMP/MPI. Along the way, multi-threaded debugging tools were used to find and fix race conditions, and performance tools were used to find and fix load imbalances.

Tiffany Broadnax hails from Jackson State University. Her summer project concentrated on Automated Performance Analysis involving the Kit for Objective Judgement and Knowledge-based Detection of Performance Bottlenecks (KOJAK) tool suite on the IBM POWER4 (MARCELLUS) at the Naval Oceanographic Office Major Shared Resource Center (NAVO MSRC). KOJAK is a set of generic tool components, including EPILOG, EXPERT, OPARI, and

CUBE, designed for the performance analysis of parallel applications, and is under consideration as part of the Well-Documented Consistent Computational Environment.

Testing and evaluation of these tools were done by doing a performance analysis of the HYbrid Coordinate Ocean Model (HYCOM) application (courtesy of Dr. Allan Wallcraft), an ocean model that is widely used in the oceanographic community.

A previous study using PAPI and TAU was done by the Scalability and Performance Optimization Team (SPOT) led by Avi Purkayastha and Chona Guiang at the Texas Advanced Computing Center (TACC); their report provided useful background for this work.

Tiffany was to learn two things: how much effort is involved in using KOJAK as compared to the effort expended by the SPOT Team, and how much of a HYCOM performance gain do we get using KOJAK as compared with the gain realized by the SPOT.

Tiffany says that while her experience with KOJAK this summer was limited, the amount of knowledge gained along the way is priceless. "Since we were the first to install KOJAK on a DoD machine, we encountered some obstacles, but we passed our experiences to the developers, improving the KOJAK installation process. We achieved a successful installation and running of HYCOM, and we are currently working on producing an event tracefile after running HYCOM through KOJAK."

Of the summer intern program, Tiffany states: "Overall, my summer experience has been wonderful. The PET Summer Intern Program was very useful to me, and not only for a possible future in HPC. I am proud to say that I learned about several aspects of HPC along with getting a

PET Summer Interns....cont.

substantial amount of work done on my designated project. Information about topics such as load imbalance and performance analysis has been helpful with problems that we encountered this summer. As a senior Computer Science (CS) major at Jackson State University, everything we learned this summer will be useful with upcoming projects throughout the school term and later on in my career. I would highly recommend this summer intern program to anyone who is interested in a future in High Performance Computing."

Darin Woods, from Central State University, spent his summer working with a memory leak checking tool called Valgrind. Darin says, "During my time as an intern for the PET program, I gained an extraordinary amount of knowledge pertaining to my field of study. I also got a taste of what it is like to be in a working environment, and how to apply the skills I learned in my college courses to problems at the work place."

Darin further states, "Learning how to utilize the HPC machines (connecting and transferring files with Kerberos, using a queuing system) as well as how to use Linux and Linux software was quite challenging. I also had a chance to see some of the other facilities on base, including a shuttle engine test and tours of the NAVO and CHL visualization labs."

As part of his project, Darin was introduced to Perl after I had mentioned using it for running commands in a Linux environment. Darin quickly noticed how simple it was to write powerful programs using Perl that would otherwise take a large amount of time to write in any other language. For example, in C/C++, you must declare any variables that you use (integer, char, etc.), but with Perl, you need only use a single character ($ for scalar, @ for arrays), and Perl knows how to handle it. He wrote a few small Perl programs and sensed the difference between programming the same commands in C/C++.

To prepare for his work with Valgrind, Darin read a couple of articles about what it is, some of the uses for it, and how to interpret the output.

Valgrind is a General Public Licensed (GPL) system for debugging and profiling x86-Linux programs. Some of the

reasons for using Valgrind are it works with programs written in any language; it debugs and profiles the entire program, even libraries called by the program and not compiled with Valgrind; it's free; and the HPC centers that are purchasing x86-Linux clusters need high-quality debugging and profiling tools. Valgrind is one of the tools being considered as part of the Well-Documented Consistent Computational Environment.

The Valgrind installation process at University of Tennessee at Knoxville (UTK) required some extra coaxing in order to finish properly, but all went smoothly after making some configuration changes.

There were a few problems installing and running our first test code (pfm_Loader, courtesy of Dr. M.J. Miller) due to an apparent incompatibility with the Local Area Multicomputer (LAM)/MPI library, so a decision was made to install and use MPICH, an open-source multiple-platform MPI implementation.

After everything was in place, Darin ran Valgrind with default options; no errors were reported with pfm_Loader.

Next he tried the "eak-check" option which checks for memory leaks in the program; this produced no errors. Then he tried the "check-children" option; this option checks other programs and libraries accessed by the original program and produced numerous errors from the "lib" library. Finally, the "gdb-attach" utility (incompatible with the "check-children" option) opens the gdb debugger anytime an error is found in Valgrind.

The GNU Debugger (GDB) lets you debug a program while it is running and can also let you know what it was doing before it crashed. When an error is found, Valgrind stops and lets you use the GDB utility to easily correct an error without having to re-run the Valgrind tool; it does this for each error until Valgrind completes the test. Since this particular program had no memory leaks and was error free, Darin moved on to the next program.

Darin's next task was to take an example quicksort program from the University of Oregon's Tuning and Analysis Utilities (TAU), remove the TAU calls, and compile and run the program using Valgrind.

Left: Tom Cortese, Innovative Computing Laboratory (ICL)/The University of Tennessee, PET Computational Environments Onsite. Right: Tiffany Broadnax, Intern, Jackson State.

Two errors dealing with uninitialized variables were reported, but no memory allocation errors, so it was decided to purposely introduce memory errors by changing the amount of memory allocated to an array and then run Valgrind to see how good it is at finding memory errors.

Valgrind found the errors, but displayed output that gave no line numbers and had cryptic file names. We probably could have used the GDB tool to find out exactly where the problem occurred, but more research on how to read Valgrind's output is needed.

Of his experience in the program, Darin says, "I learned quite a bit during my summer, and will continue to learn more about some of the concepts we touched upon (most notably Perl). All of the people I met here were nice and supported our willingness to gain as much knowledge about our temporary jobs along with the additional material that we researched.

Overall, I'm glad I was able to participate in the PET summer internship; the experience I gained and the information I learned will help me in the future."

Fiseha Negasi, also from Central State University, says, "I enjoyed learning about High Performance Computing during my summer at Stennis Space Center. I was able to learn about different kinds of parallel programs and machines."

Fiseha went on to say, "In addition to my summer research project investigating adjoint compilers, I learned how to use MPI and OpenMP both in an interactive and a batch environment. Also, I was learning from my mentor Dr. Tom Cortese, by lectures he gave us, about topics such as MPI, OpenMP static and dynamic loop scheduling, how to solve differential equations numerically, the Mandelbrot Set, and HPC Architecture, including memory structure and cache."

His research project was to evaluate an adjoint compiler tool called ADIFOR (Automatic DIfferentiation of FORtran) for possible use within the DoD.

ADIFOR is a source transformation tool that accepts Fortran 77 code for the computation of a function and writes Fortran 77 code for the computation of the derivative. ADIFOR is being developed for use in meteorology and oceanography in the creation of adjoint models.

After learning about ADIFOR by reading research papers, Fiseha spent time with me installing and testing the ADIFOR package. Since we were using the newest beta version of the software, we encountered installation problems along the way, but as a result of our experiences the ADIFOR distribution is now much improved.

Fiseha liked his experience with the PET program and says, "I received knowledge of different kinds of programs and techniques from Dr. Tom and the time he spent with me to show some programming techniques. I have the confidence now to apply for a job anywhere or anyplace without any problems because I have received knowledge and computer skills from PET with different programming and computer systems techniques. I will encourage students to participate in the PET summer intern program as a means of receiving knowledge and of getting job opportunities for the future."

Being a mentor in the PET summer intern program has been a challenging but rewarding experience. In addition to the expected technical and administrative challenges involved with working in a high-performance computing environment, it was also important to keep in mind that the interns come from different backgrounds and have different levels of experience.

I enjoyed working with the interns, and I found it to be a valuable learning experience as well. Since most of my parallel programming experience has involved OpenMP, MPI, and Fortran, I had never developed a hybrid OpenMP/MPI program in C before, but this is the approach I took this summer since the interns had little or no experience with Fortran.

Starting from a relatively simple mathematical equation and building it into a complex parallel program was not only an ideal way to gradually introduce new concepts in programming and computer architecture, but also a lot of fun, as we could actually see improvements as each stage was completed.

As a result, I would encourage others to consider being summer intern mentors—there is some effort involved, but the rewards are worth it, and we get the rare opportunity to help build the HPC workforce of the future.

Left: Fiseha Negasi, Intern, Central State (Ohio). Right: Darin Woods, Intern, Central State (Ohio).

AIX WorkLoad Manager Running on HABU and ROMULUS

Sheila Carbonette, NAVO MSRC User Support

The Naval Oceanographic Office Major Shared Resource Center (NAVO MSRC) is currently running the LoadLeveler queuing system to manage serial and parallel batch jobs on the IBM P3, HABU, and the IBM POWERr4+, ROMULUS. In addition to LoadLeveler, the AIX WorkLoad Manager (WLM) has been turned on to help control resource utilization during periods of peak system demand.

The WLM monitors system resources and regulates their allocation to processes running in batch. These actions prevent jobs from interfering with each other when there are conflicting resource requirements.

When developing LoadLeveler batch scripts, Users are required to specify the "resources" directive. This directive is used to set the "ConsumableCpus" and "ConsumableMemory" LoadLeveler variables like this:

```
#@ resources = ConsumableCpus(N)
ConsumableMemory(M mb)
```

Where (N) refers to the number of Central Processing Units (CPUs) required per task for parallel jobs or per node if the job is serial, and (M mb) refers to the amount of physical memory space in megabytes required per task. For OpenMP jobs that are running as one task, this would be the total amount of memory needed for the application. For Message Passing Interface (MPI) jobs, this is the amount of memory needed by each task.

When specifying resources in a LoadLeveler batch script, it is important to remember that all the resource requests are listed on one line. If more than one line is used, the previous line(s) are overwritten.

What follows are LoadLeveler example batch scripts for submitting OpenMP, MPI, and serial jobs along with a brief explanation on the resource directives.

An example of a LoadLevler script for an OpenMP program:

```
#/bin/csh
#@ job_name       = myopenmp
#@ output         = $(job_name).o$(jobid)
#@ error          = $(job_name).o$(jobid)
#@ account_no     = Project_Name
#@ wall_clock_limit = 5:00
```

```
#@ job_type       = parallel
#@ node_usage     = not_shared
#@ node           = 1
#@ tasks_per_node = 1
#@ resources      = ConsumableCpus(16)
ConsumableMemory(8192 MB)
#@ class          = batch
#@ queue
# start job
setenv OMP_NUM_THREADS 16
setenv WORK_DIR /scr/shecar
poe $WORK_DIR/myopenmp
# end of job
```

In the previous example "ConsumableCpus" is set to 16, which matches the value for the "OMP_NUM_THREADS" environment variable. This is used to specify the number of CPUs per task. "ConsumableMemory" is set to 8192 megabytes (MB) or 8 gigabytes (GB) to allow approximately 512 MB per task.

An example of a LoadLeveler script for a serial job:

```
#/bin/csh
#@ job_name       = serialrun
#@ output         = $(job_name).o$(jobid)
#@ error          = $(job_name).o$(jobid)
#@ account_no     = Project_Name
#@ wall_clock_limit = 5:00
#@ job_type       = serial
#@ node_usage     = not_shared
#@ node           = 1
#@ tasks_per_node = 1
#@ resources      = ConsumableCpus(1)
```

```
ConsumableMemory(96 MB)

#@ class     = batch

#@ queue

# start job

setenv WORK_DIR /scr/shecar

$WORK_DIR/serialrun.exe

# end of job
```

In the previous example, "ConsumableCpus" is set to 1, and "ConsumableMemory" is set to 96 MB. This job requires one CPU and 96 MB of memory.

An example of a LoadLeveler script for an MPI job:

```
#/bin/csh

#@ environment   = ENVIRONMENT
=BATCH; COPY_ALL

#@ job_name      = mpirun

#@ output  = $(job_name).o$(jobid)

#@ error   = $(job_name).o$(jobid)

#@ account_no    = Project_Name

#@ wall_clock_limit = 2:00:00

#@ job_type      = parallel

#@ node_usage    = not_shared

#@ network.MPI   = csss,shared,US

#@ node          = 2

#@ tasks_per_node = 4

#@ resources     = ConsumableCpus(1)
ConsumableMemory(832 MB)

#@ class         = batch

#@ queue

# start job

setenv WORK_DIR /scr/shecar

poe -nodes 2 $WORK_DIR/mpihello

# end of job
```

In the previous example, "ConsumableCpus" is set to 1 for each MPI task. "ConsumableMemory" is set to 832 MB to allow 832 MB per MPI task. Users are cautioned that the amount of memory specified for "ConsumableMemory" is the amount of memory per MPI task. If the number of tasks multiplied by "ConsumableMemory" is more than the amount of

memory available on a node, then the job will not run. The LoadLeveler command, "llstatus" with the "-R" option can be used to generate a listing of all consumable resources available on each node. For example:

```
habu% llstatus -R

Machine     Consumable Resource(Available,
Total)

-------------------- ---------------------------------

h10n10s.navo.hpc.mil   ConsumableCpus(0,4)
ConsumableMemory(0.000 mb,3.418 gb)

h10n11s.navo.hpc.mil   ConsumableCpus(0,4)
ConsumableMemory(300.000 mb,3.418 gb)

h10n12s.navo.hpc.mil   ConsumableCpus(0,4)
ConsumableMemory(0.000 mb,3.418 gb)


romulus% llstatus -R

Machine     Consumable Resource(Available,
Total)

-------------------- ---------------------------------

r1n10s.navo.hpc.mil   ConsumableCpus(0,8)
ConsumableMemory(0.000 mb,6.500 gb)

r1n11s.navo.hpc.mil   ConsumableCpus(0,8)
ConsumableMemory(0.000 mb,6.500 gb)

r1n12s.navo.hpc.mil   ConsumableCpus(0,8)
ConsumableMemory(0.000 mb,6.500 gb)
```

Networking the NAVO MSRC

Randy Becnel, NAVO MSRC Network Support

The High Performance Computing (HPC) and file server platforms that comprise the Naval Oceanographic Office Major Shared Resource Center (NAVO MSRC) require high bandwidth, low latency, and high availability networks to provide connectivity among platforms and users throughout the Naval Oceanographic Office (NAVOCEANO), Naval Research Laboratory, Stennis Space Center (NRL-SSC), Defense Research and Engineering Network (DREN), and general Internet communities.

The NAVO MSRC network core is built on three Cisco 6500 series router/switches providing a triangulated, fault-tolerant 10-gigabit (Gb) Ethernet backbone among the NAVO MSRC DREN connection point, its computational resources, and the Remote Storage Facility (RSF). These high-end Layer 2/Layer 3 switches have a backplane scalable to 720 gigabits per second (Gbps) and support 10/100 BaseT, Gb and 10-Gb Ethernet interfaces. This scalability positions the NAVO MSRC to meet current and future high performance networking requirements. Connectivity to the DREN network and Internet is via an Asynchronous Transfer Mode (ATM) OC-12 (622 Megabits per second (Mbps)) Wide Area Network (WAN) link. An upgrade to an OC-48 (2.4 Gbps) is scheduled for September 2004.

High speed data transfer between computational and mass storage servers is accomplished via multiple Gb Ethernet interfaces on the host systems across a 10-Gb Ethernet backbone. User access to the computational and mass storage resources of the NAVO MSRC is provided via Gb Ethernet connectivity. High speed connectivity to a legacy Cray SV1ex cluster is provided via High Performance Parallel Interface (HiPPI) (800 Mbps) to the Cray mainframes. A HiPPI to Gb Ethernet router is then utilized to connect to the core gigabit network infrastructure.

Distribution layer connectivity consists of multiple Cisco 4507R Layer 2/Layer 3 network switches providing 10/100 BaseT and Gb Ethernet connectivity for support analyst and visualization workstations. The network switches are linked to the NAVO MSRC core infrastructure via multiple Gb Ethernet trunk links providing high speed access and fault tolerance.

NAVO MSRC network engineers manage the enterprise with the aid of the CiscoWorks 2000 Local Area Network (LAN) Management Solution software in conjunction with Concord eHealth Enterprise Management Suite. Together these enterprise management packages provide real-time network and host performance data as well as trend-analysis and capacity planning capabilities. Thoughtful planning and design of the network infrastructure and proactive monitoring of network performance has allowed the NAVO MSRC to achieve greater than 99.9 percent LAN availability to users of the NAVO MSRC resources during calendar year 2003 and the first six months of 2004.
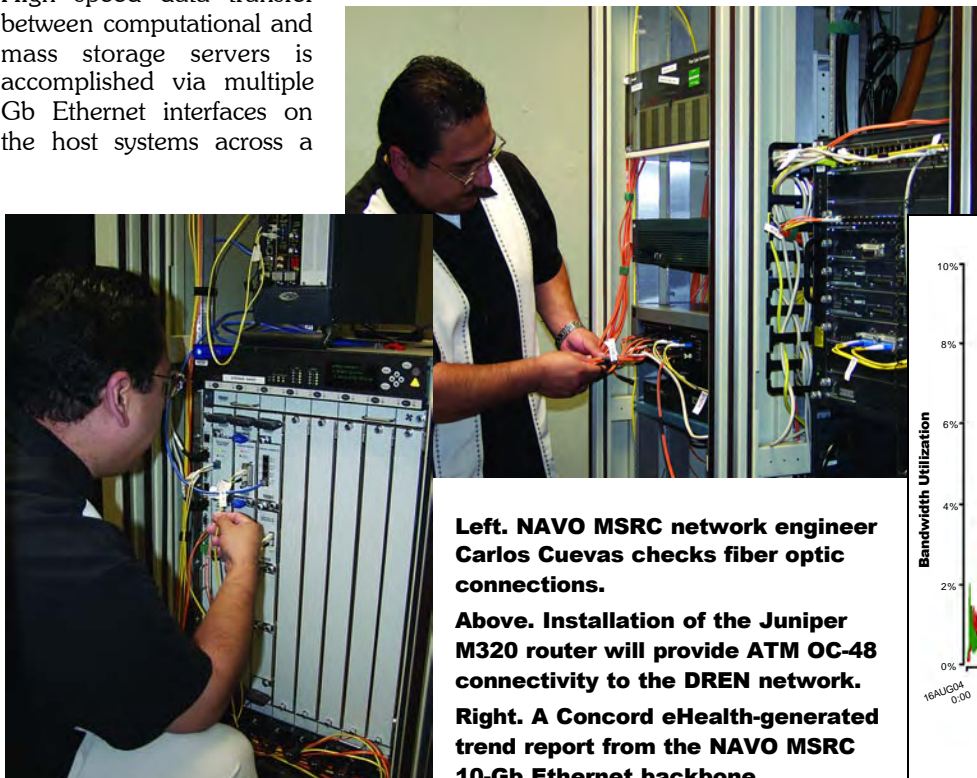
Left. NAVO MSRC network engineer Carlos Cuevas checks fiber optic connections.

Above. Installation of the Juniper M320 router will provide ATM OC-48 connectivity to the DREN network.

Right. A Concord eHealth-generated trend report from the NAVO MSRC 10-Gb Ethernet backbone.
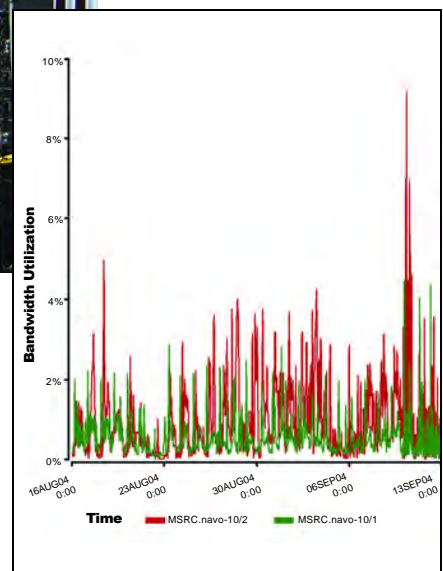
Coming Events

|	
1 - 4 **November** **2004**	**ICDM '04: The Fourth IEEE** **International Conference on Data Mining** **http://icdm04.cs.uni-dortmund.de** **Brighton, UK**
19 - 22 **December** **2004**	**HiPC 2004: The 11th Annual International** **Conference on High Performance Computing** **www.hipc.org** **Bangalore, India**
12 - 15 **February** **2005**	**SIAM Conference on Computational Science & Engineering** **www.siam.org/meetings/CSE05/** **Orlando, Florida**
12 - 16 **February** **2005**	**HPCA 2005: 11th International Symposium on High-Performance** **Computer Architecture** **http://www.hpcaconf.org/hpca11/** **San Francisco, CA**
4 - 8 **April** **2005**	**IPDPS 2005: The 19th IEEE International** **Parallel and Distributed Processing Symposium** **www.ipdps.org** **Denver, CO**
21 - 23 **April** **2005**	**Fifth SIAM International Conference on Data Mining** **www.siam.org/meetings/sdm05/** **Newport Beach, CA**
2 - 4 **May** **2005**	**NSDI '05: 2nd Symposium on Networked Systems Design and Implementation** **www.usenix.org/events/nsdi05/** **Boston, MA**
22 - 24 **June** **2005**	**ICS '05: ACM International Conference on Supercomputing** **www.ecse.monash.edu.au/conferences/ics/2005/** **Boston, MA**
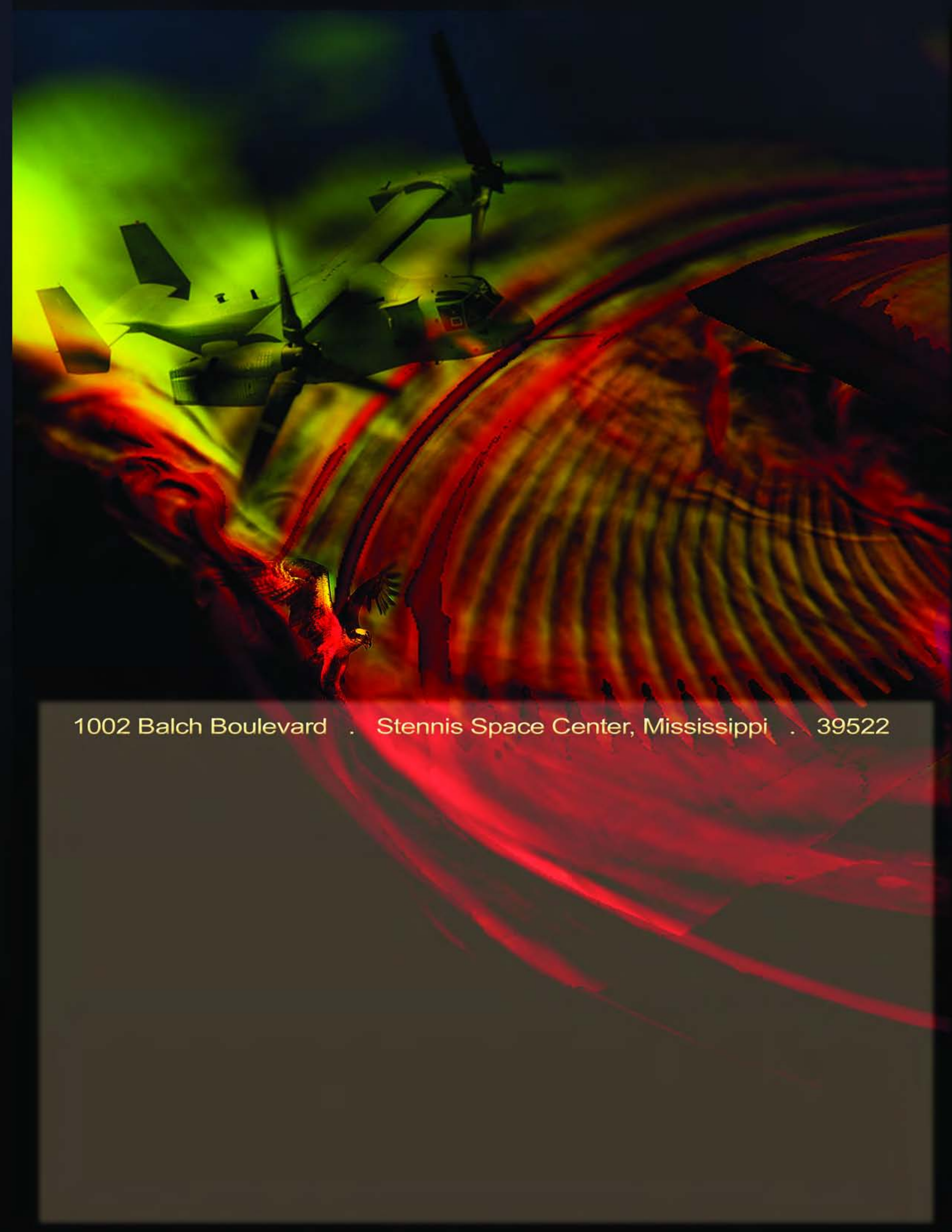
1002 Balch Boulevard . Stennis Space Center, Mississippi . 39522